

Web Modelling Languages: The Gap Between Requirements And Current Exemplars

Alice Gu, University of Technology, Sydney [\[HREF1\]](#). agu@westpac.com.au

Brian Henderson-Sellers [\[HREF2\]](#), Faculty of Information Technology [\[HREF3\]](#), University of Technology, Sydney [\[HREF1\]](#). brian@it.uts.edu.au

David Lowe [\[HREF4\]](#), Faculty of Engineering [\[HREF5\]](#), University of Technology, Sydney [\[HREF1\]](#). david.lowe@uts.edu.au

Abstract

For the development of Web systems to be effective, we need to consider *two* aspects of technical architecture: the information architecture and the functional architecture. Web modelling languages, used in the development of these systems need to be able to capture and communicate both these aspects. Specifically, a WML must provide sophisticated functions to facilitate the understanding, specification, documentation, visualisation, communication and construction of important architectural, detailed design and implementation decisions. In this paper, we identify the requirements for WMLs for these elements and, using this proposed requirements framework, evaluate six of the leading WMLs. We conclude that a number of gaps exist in existing modelling approaches, particular with regard to the level of support for modelling of: the link between information and functional designs, primarily at the architectural level; abstractions, particularly with regard to linking detailed information designs to information architectures; and the connections between business models and the system designs.

Introduction

For the development of Web systems to be effective, we need to consider two aspects of technical architecture: the information architecture and the functional architecture. These both cut across the various standard architectural elements, and must also support the Business Model. A key element of the ability to develop these systems is the availability of an appropriate web modelling language (WML^{note}). A WML must provide sophisticated functions to facilitate the understanding, specification, documentation, visualisation, communication and construction of important architectural, detailed design and implementation decisions during the course of Web system development, maintenance and evolution.

In this paper, we identify the requirements for WMLs for these two architectural elements and, using this proposed requirements framework, evaluate six of the leading WMLs.

These two groups of requirements (i.e. related to functional and informational aspects) are discussed in Sections 2 and 3, together with a section (4) on more general requirements, as a prelude to consideration of how these requirements need to be reflected in any WML. Then, in Section 5, we evaluate six of the most prominent, current WMLs using a gap analysis i.e. the difference between the goals (the requirements) and the potential achievement of those goals (current WMLs). This gap analysis then leads us to suggest future extensions that would benefit WMLs.

Web Modelling Languages - Functional Architectural Requirements

The functional characteristics of Web systems impose a number of requirements on the languages used to represent this functionality. Many of these requirements relate to the specific nature of the architecture of these systems. In the following subsections we shall explore these various requirements.

Ability To Model Integration And Connectivity

It is crucial for modelling languages to be able to represent the integration of (sub)systems and resources at high architectural levels:

- In many organisations, the newly developed Web system needs to work closely together with the organisations' existing core-business systems. These systems, the majority of which are back-end processing systems, typically reside on various operating platforms and have been developed in different programming languages. The challenge of integrating the Web systems with these legacy systems in a seamless manner therefore raises specific requirements for modelling language support.
- Since the integration of components is largely dependent on the interface specifications, WMLs need to provide the ability to model and document component interfaces accurately and unambiguously.
- It is desirable for WMLs to demonstrate the connectivity and navigability from component interface definitions.
- As the front-end interface to the organisations' core knowledge and competency, Web systems need to connect to a variety of sources of information and services, both within and beyond the organisational boundary. Representation of the connectivity and access mechanisms needs to be supported by modelling languages from a technological

perspective.

- One consequence for component interface specification is that WMLs need to ensure that component interfaces are defined consistently at all abstraction levels ([Atkinson, 1999](#)).

Ability To Support Pattern Modelling

Both research and practical experience suggest that it is desirable to use patterns whenever possible in a well structured object-oriented software system ([Gamma et al., 1995](#)). As formats for capturing, storing, distributing and reusing experience and best practices, the potential benefits of patterns include:

- Provision of guidance to software development, so that correct design decisions are made with the least possible effort. This can help to meet the tight timeframe of Web development projects.
- Because appropriate patterns are proven and commonly accepted, they usually meet a high-quality standard. The usage of patterns during software development can help to increase the total quality of the Web systems.
- Since patterns are normally shared among software developers, especially in similar application domains, the usage of patterns can actually increase the openness and interconnectivity of the Web systems.

WMLs therefore need to support the modelling of patterns, idioms, mechanisms and frameworks in various domains. One desirable feature for a WML is to be able to identify the variation points in frameworks and the rules regarding when they are applied to certain scenarios ([Fontoura et al., 2000](#)).

Ability To Represent Concepts In A Technology Neutral Fashion

Since the technologies involved in Web systems change so rapidly, it is both impossible and unwise to design a system architecture in a technology specific manner; indeed, technology-neutral design is very much needed in Web system development. As a consequence, WMLs need to provide the ability to represent these architectural and design concepts in a technology-neutral fashion.

Ability To Model Sophisticated System Functionality

Apart from simple Web sites, Web systems are also derived from conventional transaction processing systems. These systems support critical business processes and workflows that are an important part of the organisation's core business model. These crucial business functions are increasingly being implemented either through Web interfaces to existing legacy applications, or through new Web-based systems. WMLs, therefore, need to provide the ability to represent these functions and their related design artefacts. Examples include the modelling of business workflows, information and order tracking, transaction processing etc.

To maintain system integrity, WMLs should also provide the ability to define business rules and constraints that are used to ensure that both the entire system and its components have correctness, consistency and referential integrity.

Web Modelling Languages - Information Architectural Requirements

Web information architectures typically cover aspects such as content and how it is managed, information structuring and access; user contextualisation; design of, and support for, navigation; information viewpoints and presentation issues. They are usually more complex than for conventional IT systems, partly as a consequence of the heritage of these systems - evolving out of the early Web, which was primarily a distributed document management system utilising hypertext concepts to support information location and retrieval.

Ability To Model Presentation Level Concepts

Compared with conventional software systems, Web system presentational design has its own unique characteristics, which include:

- more thorough and sophisticated functions at the presentation level - a consequence of the necessarily increased quality of user interfaces;
- various media types at the presentation level, such as text, graphic, video and audio;
- WMLs will be used by not only IT professionals, but also by, e.g., graphical designers, multimedia producers, authors and market analysts.

WMLs need to support modelling in a way that is consistent with these unique characteristics.

Ability To Model Navigational Structure And Behaviour

As one of the most unique and ever-evolving features of Web systems, navigational structure and behaviour is still to be explored. However, research from hypermedia ([Schwabe and Rossi, 1998](#); [Lowe](#)

[and Hall, 1999](#); [Guell et al., 2000](#); [Koch et al., 2000](#)) supplies a more thorough understanding and better support in this area than alternatives from the software engineering field, and thus can be used as a solid foundation for further research work.

Navigation in the Web system can be either contextual or non-contextual. Whilst they both need to be supported by a well-structured Web information ontology, contextual navigation requires more consideration, such as context-related rules and constraints as well as information passing along the navigational network.

Ability To Model User Interactions With Information

Compared with conventional software systems, Web systems tend to provide much more sophisticated interactions with their users. As a consequence, WMLs need to provide the ability to model these different interactions in a complete, unambiguous and yet flexible way so that the interactions can be captured, understood, designed, implemented and maintained. We now discuss some of the relevant characteristics.

The reach of the multiple information sources

The various information sources to which a Web system needs to connect may include, but are not restricted to, databases, file servers, document repositories, internal or external Web pages, as well as other applications such as email systems and business processes. WMLs need to be able to model different information sources and their associated communication mechanisms.

The mode of access

Web systems can interact with the various information sources in either a reactive mode or a proactive model. In a reactive mode, the access action is initiated by the user and executed by the Web system, while, in a proactive mode the access action is automatically initiated by the Web system itself. An example of the latter is the maintenance of an index or taxonomy. WMLs need to provide the ability to differentiate between the two access modes and to model them accordingly.

The distribution of information

The distribution of information in Web systems involves several aspects. Firstly, it needs to be supported by the technical architecture that supports constant and reliable connectivity. Secondly, it requires the support of user profile definition so that information can be delivered to the target user or user group. Furthermore, it depends on the presentation layer to provide the ability to present the information to the user in an effective and friendly fashion. Lastly, but by no means least, it needs a disciplined security control to ensure system safety and information privacy. The involvement of all these aspects, and especially the connection between them, needs to be fully supported by WMLs.

Ability To Model User Roles And User Groups

The success of Web systems is largely dependent on user satisfaction, which is achieved by, for example, providing sophisticated functionality, easy-to-use interface and well-structured navigational architectures. One critical technique to leverage all these desirable features together and deliver them to the right people at the right time in the right format is through personalisation. WMLs must therefore provide capabilities, including modelling the definition and maintenance of user profiles, the definition of user groups, connection between users and user groups, and the linkage between user profiles/groups and user interactions.

Ability To Model Content

The effective management of content is a critical success factor of Web system development ([Lowe and Henderson-Sellers, 2001](#)). Content management usually involves, but is not restricted to, the design of the content structure, mostly expressed in the format of database schemata, document hierarchies and hyperlinks. A potential trend is to move towards an internationally accepted standard of content structure definition. Current trends suggest that XML is a likely standard, at least for the foreseeable future.

Web Modelling Languages - General Requirements

Apart from support for Web system functional architecture modelling and information architecture modelling, it is also critical for WMLs to provide some generic capabilities, such as linking the functional and information aspects of Web system technical architecture in a consistent and cohesive fashion. These generic requirements are discussed in the following sections.

Ability To Model Business Domain Concepts

Research and practical experience suggest that there is a much stronger linkage between the business model and the technical architecture in Web systems in comparison to conventional software systems. As a consequence, the quality of the technical architecture will largely depend on the developers' understanding of the current business model and the required changes to it, in the context of the impact from Web technologies. To facilitate and document this understanding, WMLs need to provide the ability to model business domain concepts e.g. business processes, business entities, workflows, business rules, together with the roles and responsibilities of users.

Business-related development and modelling artefacts are usually created and used by developers from both IT and business backgrounds. As a result, the modelling of business domain concepts needs to be designed with the consideration of target user types, so that these model artefacts can be easily understood, communicated and modified within and across development teams and business units.

Ability To Link Business Model With Technical Architecture

Once understood and documented, the business model needs to be effectively translated into a technical architecture so that the desired system functionality can be implemented and delivered. To support this requirement, WMLs need to provide the ability to identify the linkage between the business model and the technical architecture, and between the model elements in the business model and the model elements in the technical architecture. This interconnection needs to be represented at various abstraction levels.

Ability To Link Information Architecture With Functional Architecture

The requirements for WMLs have two specific foci: support for the Web information architecture and support for the Web functional architecture. It is critical that these are linked in order for the Web system to successfully address business needs in an effective way ([Lowe and Henderson-Sellers, 2001](#)). Indeed, the integrity and cohesion of a Web system is largely dependent on a close and yet flexible interconnection between its information architecture and its functional architecture. The requirement is that WMLs need to support not only the modelling of both information architecture and functional architecture but, more importantly, the integration of them in a cohesive and consistent manner.

Ability To Maintain System Integrity

The integrity of Web systems can be challenged by:

- The complexity of large scale, mission-critical, content and function-rich Web systems;
- The changes to business requirements caused by client uncertainty;
- The rapid changes of technologies;
- The continuous evolution of system functionality and structure; and
- The combination of multiple disciplines and technological areas.

However, system integrity is one of the critical quality attributes that Web system development needs to achieve. WMLs can support this goal by providing semantic definitions of both the model elements and the interrelationships between them. With the help of CASE tools, integrity and referential checking can be performed automatically or semi-automatically.

Ability To Support Understanding And Communication

To facilitate communication during Web system development and maintenance, WMLs need to provide the ability to represent various aspects of the Web systems at different abstraction levels and from different viewpoints.

To represent system structure at various abstraction levels

At different stages of the Web system life cycle, models are used to support different tasks, from requirements engineering, system analysis and design, to information structuring and presentation, to component integration and hardware specification, to maintenance, and so on. To satisfy specific requirements from these various tasks, WMLs need to provide modelling artefacts at different abstraction levels. The need for various abstraction levels is also reflected in the wide variety of skill sets in the Web system development teams.

One implication of this requirement is that WMLs need not only to provide the possibility of defining models at different abstraction levels, but also to support cohesion of the entire model by providing coordinated interconnections among these abstraction levels.

To represent models from different viewpoints

With the rapid increase of complexity in Web systems, it becomes increasingly important for WMLs to

provide different modelling artefacts that support various viewpoints. By looking at the system from different specific perspectives, developers can break down a big and complex Web system into pieces or layers that are more manageable.

Some potential viewpoints of the Web system architecture include:

- Business model and business processes;
- Presentation and user interface design;
- Navigation;
- Information content management;
- Functional requirements and associated design aspects;
- Integration;
- Non-functional requirements and their impact on system design;
- Implementation level system structure; and
- Personalisation and user profile management.

The support for these various modelling viewpoints can also lead to other potential benefits. For example, since a Web system development team usually contains people from various backgrounds and with different skill sets, it can be very helpful to customise the modelling artefacts so that familiar concepts and methods can be used by people from different disciplines.

The breakdown of a Web system model into different viewpoints should not impact on the consistency, correctness and cohesion of the entire system architecture. It is also desirable that, when one aspect of the Web system architecture is changed, the potential impact of the change can be identified and highlighted in the related aspects, by defining and maintaining the interrelationship between modelling viewpoints.

Ability To Be Process Independent

To support successful and effective Web system development, conventional software development processes and methodologies cannot be used blindly without modification and enhancement. However, this is not to say that existing software engineering approaches show no value in the process of Web system development; instead, they need to be selected and adjusted to address the unique characteristics of Web systems (e.g. [Haire et al., 2001, 2002](#); [Henderson-Sellers et al., 2001](#); [Lowe and Henderson-Sellers, 2002](#)).

In the case of process engineering and process tailoring ([Henderson-Sellers, 2002b](#)), WMLs need to be able to provide the ability to tailor their modelling artefacts themselves. One implication is that to be selected and used in the customised processes, these modelling artefacts need to have some independence, whilst still maintaining the cohesion of the entire system architecture.

Ability To Support Web System Life Cycle Management

For the majority of software systems, including Web systems, the development stage is only the beginning of their life cycles. Furthermore, because Web systems demonstrate fine-grained evolution during their life cycle, maintenance plays an increasingly important role in comparison with conventional software systems. WMLs, therefore, need to facilitate the software life cycle management of Web systems. They need to produce modelling artefacts that are suitable for not only the design and development phases but also the ongoing maintenance and evolution of Web system architecture.

Gap Analysis

A number of Web Modelling Languages have emerged over the last decade. We selected six of these for detailed study and evaluation: OOHDM ([Schwabe and Rossi, 1998](#); [Schwabe et al., 2001](#); [Rossi et al., 1999, 2000](#)), WebML ([Ceri et al., 2000](#)), Koch ([Koch and Mandel, 1999](#); [Koch et al., 2000](#)), Conallen ([2000](#)), W2000 ([Baresi et al., 2001](#)) and HDM-Lite/Autoweb ([Fraternali and Paolini, 2000](#)).

Whilst it is important to recognise the strengths of each of the existing modelling approaches for Web applications, it is much more crucial to fully understand their limitations and flaws and, in turn, the implications of these limitations on the design and development of Web systems. Our research of the existing literature in the field ([Gu, 2001](#)) shows that these limitations and their implications are far from thoroughly understood and analysed by researchers and practitioners. This could have, to a large extent, caused the lack of effective modelling language support for Web system development to date.

The Requirement Checklist

In this section, we use a checklist that contains the requirements for WMLs, derived from Sections 2-4, and use it to evaluate the six selected exemplar WMLs. The evaluation is documented in Table 1.

Table 1: *Gap Analysis (Legend: Y - Yes; N - No; P - Partial)*

Requirements for Modelling Languages	OOHDM	WebML	Koch	Conallen	W2000	HDM-lite/Autoweb
Ability to Model Integration and Connectivity	N	N	N	P	P	P
Ability to Support Pattern Modelling	Y	N	N	P	P	N
Ability to Represent Concepts in a Technology Neutral Fashion	P	Y	Y	P	Y	Y
Ability to Model Sophisticated System Functionality	N	N	N	P	N	N
Ability to Model Presentation Level Concepts	Y	Y	Y	N	N	Y
Ability to Model Navigational Structure and Behaviour	Y	Y	Y	P	P	Y
Ability to Model User Interactions with Information	P	Y	Y	P	Y	Y
Ability to Model User Roles and Groups	N	Y	N	P	Y	Y
Ability to Model Content	Y	Y	Y	P	Y	Y
Ability to Model Business Domain Concepts	P	P	P	P	P	P
Ability to Link Business Model with Technical Architecture	P	P	P	P	P	P
Ability to Link Information Architecture with Functional Architecture	N	N	N	N	P	N
Ability to Maintain System Integrity	N	P	N	N	P	P
Ability to Support Understanding and Communication	Y	Y	P	P	Y	Y
Ability to be Process Independent	N	Y	Y	Y	N	N
Ability to Support Web System Life Cycle Management	P	P	P	P	P	P

From the patterns shown in this checklist, some major limitations of existing modelling approaches can be readily identified. These limitations are discussed in further detail below.

The Limitations

Inability to model sophisticated functionality

Most existing WMLs largely fail to address the modelling of sophisticated functionality found in Web systems. The reasons may include, but are not restricted to:

- For approaches from a hypermedia background, the focus has been on the organisation, presentation and navigation of information. Functionality, especially core business functions such as mission-critical transactions, have consistently been overlooked;
- For approaches that adapt traditional object-oriented modelling languages, a lack of appreciation of Web system specific functionality can be observed in these applications and extensions. They normally tend to model Web systems in a similar manner to conventional software systems, or simply define some extensions at the implementation level.

Disconnection between functional architecture and information architecture

As Web systems have evolved from information repositories to distributed applications, the requirement to blend navigation and browsing capabilities with operations and transactions increases. Web systems can be viewed as ([Baresi et al., 2000](#)):

- An extension to the traditional information systems, complemented with navigation and complex information structures; or
- An extension to traditional Web sites complemented with various operations and conventional functionality of information systems.

Either way, the coupling between hypermedia information structures and operational features is important, but often overlooked. The majority of existing approaches have a hypermedia origin (e.g. OOHDM, WebML, Koch, W2000 and HDM-lite). They usually provide reasonably rich and sophisticated features that support the Web system information architecture, in particular the modelling of navigation and presentation. On the other hand, they offer very limited modelling capabilities to facilitate complex functional designs of Web systems.

In contrast, Conallen's approach aims to extend the UML notation for supporting the modelling of Web system operations and interactions, whilst the information architecture aspect of Web systems is inadequately addressed.

In summary, due to the lack of support for both the functional architecture and the information architecture, existing modelling approaches cannot provide the capability to define a cohesive interconnection between these two integral parts of the Web system technical architecture.

Disconnection between business model and technical architecture

There is an obvious sign of disconnection between the business model and the technical architecture in the existing modelling approaches for Web applications. The potential reasons may include:

- Most of the approaches aim to facilitate the design and implementation of software systems, and requirements engineering and business modelling are not a focus for the majority of the approaches discussed here.
- Due to the unique characteristics of Web systems, including the uncertainty in the business domain and the rapidly changing user requirements, there is no commonly accepted and well-proven approach that can effectively capture the business requirements and translate them into technical architecture during Web system design. Although there has been research work that focuses in this area ([Lowe and Eklund, 2001](#); [Lowe and Elliott, 2001](#)), an ultimate solution is yet to be identified.

A major reason that most of the studied approaches fail to address this issue properly is that they are unable to maintain a balance between the support for the information architecture and the functional architecture as discussed above. As a result, whilst some approaches manage to translate a business model into an information architecture but leave functional aspects unaddressed, others tend to demonstrate the converse.

Potential misuse of uml extension mechanisms

Some of the modelling approaches, including the Koch approach and Conallen's approach, are extensions of existing UML notations. Apart from their inability to offer desirable capabilities for Web system modelling as studied above, they also tend to demonstrate potential issues related to modelling language extensions, such as the improper usage of stereotypes ([Henderson-Sellers, 2002a](#)).

Inability to support system life cycle management

In response to the ever-changing nature of Web systems and the increasing emphasis on system maintenance, WMLs need to provide adequate capability to support system life cycle management, including documentation of all work products.

Most of the existing approaches studied concentrate on providing notations for the design and implementation phases of Web system development, while other stages such as requirements engineering and ongoing maintenance are not properly supported.

Support Summary

As discussed above, several gaps were identified during the analysis of the existing modelling approaches for Web applications. In Figure 1, we translate the results of the gap analysis into a visual representation in terms of WML support for functional and informational architecture modelling. This diagram was created by translating the data in Table 1 to numerical values (assigning 1, 0 and 0.5 to Yes, No, and Partial respectively) and then summing the results. We have also added three additional notations to the diagram for comparison (the Unified Modelling Language, the Hypermedia Design Model, and the Relationship Management Model - the first is a mainstream software modelling language, and the latter two are early hypertext modelling languages), though their positions are only indicative as they have not been formally determined in the same way as the other languages. Although somewhat crude, Figure 1 demonstrates that most existing modelling approaches fail to provide concurrent support for both information architecture and functional architecture design.

Existing Modelling Approach Gap Analysis

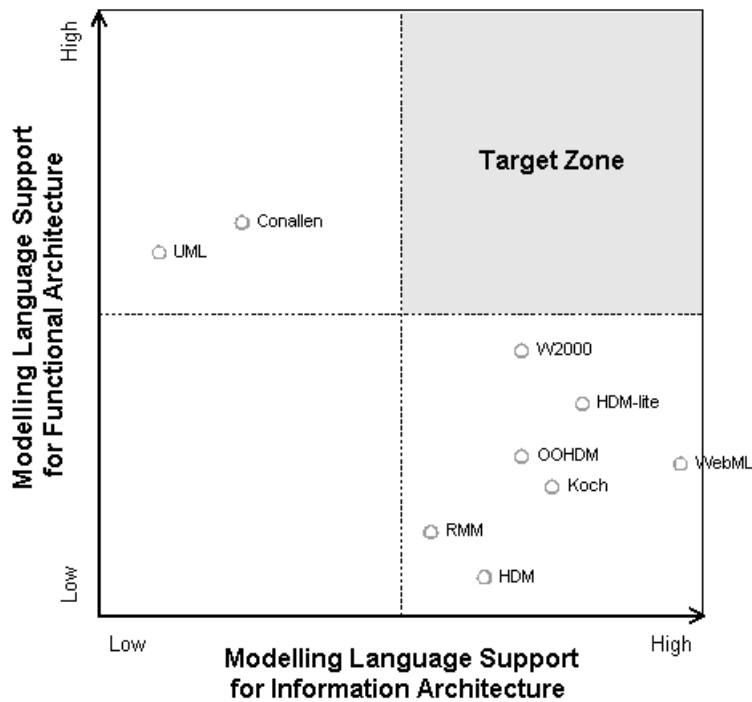


Figure 1: *Existing Modelling Approach Gap Analysis*. In this diagram we have also included, for comparison purposes only, three additional notations: UML, HDM and RMM.

Summary and Future Work

In this paper, we have explored the requirements on web modelling languages for representing the characteristics of Web systems and where existing languages fail to meet these requirements. Existing limitations tend to be most significant in several key areas: the lack of support for effective modelling of the link between information and functional designs, primarily at the architectural level; the lack of modelling abstractions, particularly with regard to linking detailed information designs to information architectures; and unclear connections between business models and the system designs. Ongoing work (to be published elsewhere) will describe our proposed extensions to Web Modelling Languages that address, in part, some of these issues.

The key message that arises from this work is that these apparent gaps in the modelling languages for Web applications are effectively restricting our ability to model, and therefore analyse and understand, key aspects of Web system design. Most noticeable in this area is the impact of the disconnection between functional and informational models on our ability to construct systems that allow us to interact with the information in rich ways; and the impact of the disconnection between detailed design and abstract designs (Gu *et al.*, 2002) on our ability to effectively manage the evolution of Web systems whilst ensuring ongoing architectural integrity.

References

- Atkinson, C. (1999) "Supporting and Applying the UML Conceptual Framework", *The Unified Modeling Language. <> '98: Beyond the Notation*, LNCS1618, Springer-Verlag, Berlin, 21-36
- Baresi, L., Garzotto, F. and Paolini, P. (2000), "From Web Sites to Web Applications: New Issues for Conceptual Modelling", *Procs. ER'00*, Springer, pp89-100.
- Baresi, L., Garzotto, G. and Paolini, P. (2001) "Extending UML for Modelling Web Applications", *Procs. 34th Hawaii International Conference on System Sciences*
- Ceri, S., Fraternali, P. and Bongio, A. (2000) "Web Modelling Language (WebML): A Modelling Language for Designing Web Sites", *Procs. WWW9/Computer Networks 33*, 137-157
- Fontoura, M., Pree, W. and Rumpe, B. (2000) "UML-F: A Modelling Language for Object-Oriented Frameworks", *Procs. ECOOP 2000*, Springer, 63-82
- Fraternali, P. and Paolini, P. (2000) "Model-Driven Development of Web Applications: the AutoWeb System", *ACM Transactions on Office Information Systems*, 18(4)

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995) *Design Patterns. Elements of Reusable Object-Oriented Software*. Addison-Wesley.

Gu, A. (2001) *Extending Object-Oriented Modelling Languages for Web Applications*, MSc Thesis, University of Technology, Sydney, 182pp

Gu, A., Lowe, D. and Henderson-Sellers, B. (2002) "Linking modelling capabilities and abstraction levels: the key to web system architecture integrity", Procs. *WWW 2002, The Eleventh International World Wide Web Conference*, 7-11 May, Hawaii, USA, published on CD-ROM and available online at [\[HREF6\]](#)

Guell, N., Schwabe, D. and Vilain, P. (2000) "Modelling Interactions and Navigation in Web Applications", Procs. *ER'00*, Springer, Salt Lake City

Haire, B., Henderson-Sellers, B. and Lowe, D. (2001) "Supporting Web Development in the OPEN Process: Additional Tasks," Procs. *COMPSAC2001* IEEE Computer Society Press, pp383-389

Haire, B., Henderson-Sellers, B. and Lowe D. (2002) "OPEN Process Support for Web Development," *Annals of Software Engineering* (in press)

Henderson-Sellers, B. (2002a) "Notes on the use of subtypes and stereotypes in the UML metamodel", *J. Database Management*, April-June. pp44-52

Henderson-Sellers, B. (2002b) "Agile or rigorous OO methodologies - getting the best of both worlds", *Cutter IT Journal*, 15(1), pp25-33

Henderson-Sellers, B., Haire, B. and Lowe D. (2001) "Adding Web Support to OPEN," *JOOP*, 14(3), pp34-38

Koch, N. and Mandel, L. (1999) *Using UML to Design Hypermedia Applications*, Technical Report 9901, Ludwig-Maximilians-University, March

Koch, N., Baumeister, H., Hennicker, R. and Mandel, L. (2000) "Extending UML to Model Navigation and Presentation in Web Applications", Procs. *Modelling Web Applications in the UML Workshop, UML '2000*, York, England, October

Lowe, D. and Eklund, J. (2001a) "Commercial Issues in Specification of Web Systems," Procs. *2001 Australian Workshop on Requirements Engineering*, November, UNSW, pp4-13

Lowe, D. and Elliott, V.V. (2002a) "Web Requirement practices: An Overview", *Journal of the American Society for Information Science and Technology (JASIST)*, (Submitted)

Lowe, D. and Hall, W. (1999) *Hypermedia & The Web: An Engineering Approach*, John Wiley & Sons, 599pp

Lowe, D. and Henderson-Sellers, B. (2001b) "Characteristics of Web Development Processes", Procs. *SSGRR 2001: International Conference on Advances in Infrastructure for Electronic Business, Science and Education on the Internet*, (ed. V. Milutinovic), Scuola Superiore Guglielmo Reiss Romoli, Italy, 21pp

Lowe D. and Henderson-Sellers B. (2002b) "Characterising Web Systems: Merging Information and Functional Architectures," Procs. *Architectural Issues of Web-Enabled Electronic Business*, S. Nansi, Ed., Idea Group Publishing

Lowe D. and Henderson-Sellers B. (2001) "Web Development: Addressing Process Differences," *Cutter IT Journal*, 14(7), pp11-17

Rossi, G., Schwabe, D. and Lyardet, F. (1999) "Web Application Models Are More Than Conceptual Models", Procs. *World Wide Web and Conceptual Modelling'99 Workshop, ER'00 Conference*, Springer, pp239-253

Rossi, G., Schwabe, D. and Lyardet, F. (2000) "Abstraction and Reuse Mechanisms in Web Application Models", Procs. *World Wide Web and Conceptual Modelling'00 Workshop, ER'00 Conference*, Springer, 76-88

Schwabe, D. and Rossi, G. (1998) "Developing Hypermedia Applications using OOHDM", Procs. *Workshop on Hypermedia Development Processes, Methods and Models (Hypertext'98)*, Pittsburgh, USA

Schwabe, D., Rossi, G., Esmeraldo, L. and Lyardet, F. (2001) "Engineering Web Applications for Reuse", *IEEE Multimedia*, Spring, pp2-12

Hypertext References

HREF1

<http://www.uts.edu.au>

HREF2

<http://www-staff.it.uts.edu.au/~brian>

HREF3

<http://www.it.uts.edu.au/>

HREF4

<http://www.eng.uts.edu.au/~dbl/>

HREF5

<http://www.eng.uts.edu.au/>

HREF6

<http://www2002.org/CDROM/poster/120.pdf>

Copyright

Alice Gu, Brian Henderson-Sellers, David Lowe, © 2002. The authors assign to Southern Cross University and other educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to Southern Cross University to publish this document in full on the World Wide Web and on CD-ROM and in printed form with the conference papers and for the document to be published on mirrors on the World Wide Web.